

**IN THE CLAIMS:**

Please amend the claims as follows:

1. (Currently Amended) A method for reducing contention in a multithreaded processor; the method comprising the steps of
  - (a) providing at least two thread stack/thread heap combinations in an address space on the processor, wherein each thread heap is for thread local memory usage and wherein each thread stack and thread heap combination ~~grew~~ grows in opposite directions; and
  - (b) providing a dead zone between the at least two thread stack/thread heap combinations, wherein data can not be written to, read from and executed from the dead zone.
2. (Cancelled)
3. (Cancelled)
4. (Previously Presented) The method of claim 1 wherein the providing step (a) further comprises the steps of:
  - (a1) providing a base address for each\_thread stack;
  - (a2) creating an initialization for each thread heap from the base address; and
  - (a3) assigning memory regions in the address space to each thread stack and thread heap combination.
5. (Previously Presented) The method of claim 4, further comprising the step of
  - (a4) allowing the assigned memory regions of each thread stack and thread heap combination to grow in opposite directions as needed.

6. (Currently Amended) A computer readable medium containing program instructions for reducing contention in a multithreaded processor; the program including instructions for:

(a) providing at least two thread stack/thread heap combinations in an address space on the processor, wherein each thread heap is for thread local memory usage and wherein the each thread stack and thread heap combination ~~grew~~ grows in opposite directions; and

(b) providing a dead zone between the at least two thread stack/thread heap combinations, wherein data can not be written to, read from and executed from the dead zone.

7. (Cancelled)

8. (Cancelled)

9. (Previously Presented) The computer readable medium of claim 6 wherein the program instructions (a); further comprise the steps of:

(a1) providing a base address for each thread stack;

(a2) creating an initialization for each thread heap from the base address; and

(a3) assigning memory regions in the address space to each thread stack and thread heap combination.

10. (Previously Presented) The computer readable medium of claim 9, wherein the instructions further comprise, instructions for (a4) allowing the assigned memory regions of each thread stack and thread heap combination to grow in opposite directions as needed.

11. (Currently Amended) A system for reducing contention in a multithreaded processor; the system comprising:

means for providing at least ~~one~~ two thread stack/thread heap combinations in an address space on the processor, wherein each thread heap is

for thread local memory usage and wherein each thread stack and thread heap combination grew grows in opposite directions,

means for providing a dead zone between the two thread stack/thread heap combinations, wherein data can not be written to, read from and executed from the dead zone.

12. (Cancelled)

13. (Cancelled)

14. (Previously Presented) The system of claim 11 wherein the means for providing at least two thread stack/thread heap combinations further comprises:

means for providing a base address for each thread stack;

means for creating an initialization for each thread heap from the base address; and

means for assigning memory regions in the address space to each thread stack and thread heap combination.

15. (Previously Presented) The method of claim 14 which includes means for allowing the assigned memory regions of each thread stack and thread heap combination to grow in opposite directions as needed.